**Naval Research Laboratory**

**Washington, DC 20375-5320**



NRL/FR/8154--93-9578

(2)

# NAVSPASUR Sensor System Digital Signal Processing Receiver

## Volume 2—Function and Capabilities of Hardware and Software Components

CARL J. MORRIS
CAROLYN F. BRYANT
MARILYN P. EARL
TAMARA A. MYERS

*Space Applications Branch*
*Space Systems Development Department*



September 30, 1993



Approved for public release; distribution unlimited.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY *(Leave Blank)* | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | September 30, 1993 | Final October 1990-July 1993 |

4. TITLE AND SUBTITLE

NAVSPASUR Sensor System Digital Signal Processing Receiver
Volume 2—Function and Capabilities of Hardware and Software Components

5. FUNDING NUMBERS

PE - 12427N
TA - X-0125
WU - DN 780-065

6. AUTHOR(S)

Carl J. Morris, Carolyn F. Bryant, Marilyn P. Earl, and Tamara A. Myers

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/8154--93-9578

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Space Surveillance Center
Dahlgren, VA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

| 12a. DISTRIBUTION/AVAILABILITY STATEMENT | 12b. DISTRIBUTION CODE |
|---|---|
| Approved for public release; distribution unlimited. | |

13. ABSTRACT *(Maximum 200 words)*

This is a system description of the Naval Space Surveillance (NAVSPASUR) Sensor System Digital Signal Processing Receiver (DSPR). The NAVSPASUR system began as an advanced research project in June 1958, was commissioned as an operational Naval command in February 1961, and is operated by the Naval Space Surveillance Center (NSSC) in Dahlgren, Virginia. The DSPR is a real-time radar data acquisition and analysis system. Its function is to detect, with no prior information, all space objects whose orbits cross the continental United States and to compute their subsequent orbits. It provides vital satellite information in support of national defense mission objectives of space intelligence, satellite attack warning, satellite intercept support, and space mission support. This system description was prepared as part of a modernization program that has replaced DSPR hardware for which parts are no longer available.

Volume 1 (NRL/FR/8154--93-9577) describes the DSPR system in terms of current operation and hardware and software environment. Functions of the major subsystems and the relationship between them are discussed.

Volume 2 (NRL/FR/8154--93-9578) discusses the function and capabilities of software and hardware components of the subsystems that provide the digital functions of the DSPR. For each subsystem, individual software modules and hardware components used primarily by that subsystem are described.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Radar Satellite Interferometry | Volume 2, 40 pages |
| | 16. PRICE CODE |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

## CONTENTS

# NAVSPASUR SENSOR SYSTEM DIGITAL SIGNAL PROCESSING RECEIVER

## Volume 2—Function and Capabilities of Hardware and Software Components

## 1. INTRODUCTION

This is Volume 2 of a four-volume system description of the Naval Space Surveillance (NAVSPASUR) Sensor System Digital Signal Processing Receiver (DSPR) hardware and software. The hardware was developed by the Naval Research Laboratory (NRL) for NAVSPASUR. The original software was designed by NRL and developed jointly by Digital Equipment Corporation (DEC) and NRL. The modernized software described in this volume was designed and developed by NRL.

The NAVSPASUR system began as an advanced research project in June 1958. In October 1960, the project was transferred from the Advanced Research Projects Agency to the Navy, and was subsequently commissioned as an operational Naval command in February 1961. Since then, it has been operated by the Naval Space Surveillance Center (NSSC) in Dahlgren, Virginia. The NSSC is responsible to the Chief of Naval Operations for support to the operating forces of the United States Navy, and is under the operational control of the U.S. Space Command, Colorado Springs, Colorado, for those space object data collection functions that are part of the National Space Detection and Tracking System (SPADATS).

### 1.1 Purpose and Scope

This volume describes the function and capabilities of the individual software and hardware components of each subsystem of the DSPR. Volume 1 presents an overview of the hardware and software of the system. Volume 3 describes the operating system functions required by the applications software. Volume 4 discusses hardware interfaces between the major subsystems of the DSPR.

The documentation is part of a project that has replaced DSPR hardware for which parts are no longer available. Hardware replaced included the DEC PDP-11/60 minicomputers that served as the central processors for the DSPR system, the Floating Point Systems Array Processors that performed fast Fourier transforms on data from the alert antennas, and various interface hardware. New hardware installed includes DEC VAX 4000 system 200 (VAX 4200) minicomputers, CSPI MAP 4000 array processors, and new interface hardware. All new hardware and associated software was required to interface with the rest of the existing system. The new software was required to replicate all the functions of the existing software.

The set of documents describes the hardware and software components of the DSPR system in terms of its current functionality. It will serve as a baseline description from which future specifications for upgrades to hardware and software may be drawn.

Manuscript approved June 14, 1993.

### 1.2 System Overview

The Digital Signal Processing Receiver is a real-time radar data acquisition and analysis system. The function of the NAVSPASUR system is to detect, with no prior information, all space objects whose orbits cross the continental United States and to compute their subsequent orbits.

NAVSPASUR is a multistatic continuous-wave radar system operating as a large radio interferometer, with nine stations located along a great-circle path across the southern United States. The inclination of the great circle is 33.57 degrees with respect to the equator.

The system consists of three transmitters and six receivers. The stations are located as follows:

<u>Transmitters</u>: Jordan Lake Station, Wetumpka, Alabama
Lake Kickapoo Station, Archer City, Texas
Gila River Station, Maricopa, Arizona

<u>Receivers</u>: Tattnall Station, Glennville, Georgia
Hawkinsville Station, Hawkinsville, Georgia (high altitude)
Silver Lake Station, Hollandale, Mississippi
Red River Station, Lewisville, Arkansas
Elephant Butte Station, Truth or Consequences, New Mexico (high altitude)
San Diego Station, Chula Vista, California

Each transmitting station radiates a continuous wave of radio energy that combines with the other transmitting stations' beams to form the NAVSPASUR "fence." When an object, such as a satellite, enters the fence, a small fraction of the radio energy is reflected to one or more of the receiver sites. The receiving stations use large multiple-array interferometers to detect the reflected signal and to measure its angle of arrival. The transmitter and receiver arrays are cross-polarized to prevent the transmitted energy from reaching the receivers without having been reflected from a space object. Each receiving station transmits phase and amplitude data, along with frequency identifiers, statistical measures, and time stamps to the NAVSPASUR Processing and Operations Center at the NSSC via a dedicated telephone line, where local direction angles for each object are computed.

The antenna data available at the receiver stations can be processed to produce three types of data: full-Doppler, half-Doppler, and quarter-Doppler These three types are also referred to as low-altitude, mid-altitude, and high-altitude, respectively. The output of the full-Doppler data is far more important than either the half- or quarter-Doppler data. For this reason, production of the full-Doppler data by the DSPR system takes priority over half- and quarter-Doppler processing. The purpose, then, of each DSPR is to provide a continuous stream of full-Doppler data to the NAVSPASUR Processing and Operations Center.

### 1.3 DSPR System Philosophy

The DSPR system philosophy stems directly from the system purpose. This system is designed to run continuously, with any single component failure resulting in, at most, a very short (less than a minute) interruption in full-Doppler data processing. For this reason, each DSPR actually consists of two systems that duplicate each other. One of these systems normally handles the full-Doppler data from the antennas, and is called the primary system. The other system normally handles the half- and quarter-Doppler data and is called the secondary system. If the primary system malfunctions and is no longer

able to process data, the secondary system detects the failure and declares a "primary system failure." The secondary system then reinitializes itself to become a primary system. This reinitialization process occurs only in the secondary system. If, under normal conditions, the secondary system malfunctions, the primary system detects the failure, but does not attempt to process the half- and quarter-Doppler data. This dual system philosophy is embedded into every aspect of the DSPR system.

### 1.4 DSPR Subsystems

For the purpose of this document, the DSPR is treated as consisting of the following subsystems: System Monitor and Control, Target Detection and Selection, Interferometer Data Collection, Data Processing, Data Line Communications, Interprocessor Communications, System Timing Components, Operational Tests, and Utility Bus Control. These subsystems provide the digital functions of the DSPR system. An RF subsystem, which converts the analog outputs from the antenna arrays to digital form for subsequent processing, is not described in this document.

Sections 2 through 10 discuss each digital subsystem in terms of software and hardware components. Under software components, the Fortran modules and VAX Macro assembly language modules for that subsystem are described. Under hardware components, the hardware used primarily by that subsystem is described. Except where noted, all components, hardware and software, are duplicated in the primary system and the secondary system. Section 11 covers miscellaneous utility subroutines that are called by routines in various subsystems. Manuals for specific hardware components are listed in References.

The descriptions in this document assume a working knowledge of VAX 4000 computers, the VAX VMS operating system, and the VAX Fortran 77 and VAX Macro programming languages.

## 2. SYSTEM MONITOR AND CONTROL

The System Monitor and Control subsystem has three major functions: initializing all other procedures, controlling the system operation, and serving as the interface between the operators and the system. The procedure that controls these functions is SYSMON and its associated subroutines. Related hardware consists of a central controller and an operator's console system.

### 2.1 Fortran Modules

*Program SYSMON*

This is the main program for the system monitor. It initializes all other procedures in the system, interfaces to commands entered through the operator's console, and coordinates the system's response to error conditions.

The start-task mechanism for each module is to create a mailbox for the task, send an initialization request packet to the mailbox, create a detached process, and start the task running in the process. The task then reads its mailbox to receive the initialization packet (which in some cases tells the task whether it is being started as part of the primary or secondary system), attempts to perform any necessary startup functions, and send back an initialization response message indicating success or failure in starting. SYSMON waits for the response packet and checks the response code. If the task has started successfully, SYSMON continues; if not, it sends a failure message to the operator.

*Subroutine ABORT_TASK (PID)*

This routine is called to abort a task with the DSPR system. The argument PID is the process identification of the task to be aborted.

*Subroutine ACKNOWL*

This routine processes the operator's acknowledgment of an error condition.

*Subroutine ALARM (JTYP)*

This routine tries to turn on the alarm. The calling argument JTYP is 4 (octal) for the visual alarm and 14 (octal) for both the visual and audible alarms.

*Subroutine CALFRQ (DOPPLR, BIN, FREQ)*

This routine converts a Doppler offset to center frequency into an ASCII encoded frequency and a bin number. Argument DOPPLR is the Doppler offset to center frequency; BIN is the frequency on which to collect, expressed as a bin number; and FREQ is the frequency encoded in ASCII format.

*Subroutine DISPATCH*

This routine prompts the system operator for a command. It waits for the command to be received, using the ICMDEF local event flag, or a time-out to occur. Once a command is received, control is dispatched to the appropriate subroutine. If a time-out occurs, the dynamic display is reactivated. Subroutines that it calls are the other subroutines that allow the operator to control the system, for example, DISPLAY displays the status of the DSPR system; EXAMINE allows the operator to examine system parameters; SCSHUT allows the operator to shut down the system.

*Subroutine DISPLAY*

This routine displays the status of the DSPR system on the primary CRT. The display is dynamically refreshed on the CRT every 10 seconds. If the operator enters any keyboard character, the CRT is cleared and control returns to routine DISPATCH. This module also functions as an activity monitor for the primary system. If, in the time period of 3 minutes, zero targets or more than 90 targets are processed, an error condition is declared.

*Subroutine DLY_MOD (DDMOD)*

This routine is used by the operator to modify the data delay parameters for those stations that have data delay. Argument DDMOD is a flag to indicate that parameters are to be modified.

*Subroutine EXAMINE (JVAL)*

This routine allows an operator to examine a group of system parameters, for example, antenna parameters, data processing parameters, etc. The routine displays a menu listing the available commands and what they do. When a command (single character) is typed into the console, the requested parameters are displayed on the console terminal and on the hardcopy device. Argument JVAL is set to 0 for manual display, in which the operator makes the selection. It is set to 1 for automatic display, in which MODxx variables are tested to decide what to display.

EXAMINE is also called from DISPATCH after certain parameters have been changed. In that case, the MODxx variables control what parameters are displayed.

*Subroutine FILTER (IVAL)*

This routine allows the operator to examine, enable, disable, or modify filtering parameters such as signal strength, Doppler tolerances, and frame duration. Argument IVAL is set to 0 for manual operation, to 1 for automatic operation.

*Subroutine GET_RESP (IERR)*

This routine waits for an initialization response from a procedure. If the response is received, it sets IERR to 1, indicating success; otherwise it sets IERR to -1.

*Subroutine INIT_ICCBUF (BUFNUM, IMTYPE, SUBTYPE)*

This module fills the send buffer for a send to the ICC procedure. It also puts the message that is to be sent to ICC in global common area DSPCOM. Argument BUFNUM is the ICC sub-buffer number; IMTYPE is the ICC message type; and SUBTYPE is the subtype for specific messages.

*Subroutine INIT_P1*

This routine initializes the permanent components of the DSPR system. The initialization is performed only once at system startup, and is not repeated during reinitialization. For initialization of the primary system, the interprocessor communications procedure is started, the processing mode is established. The antenna data collection procedures, the time stamp procedure, and the utility bus procedures are started. In addition, automatic reference calibration and system signal confirmation tests are run, and the digital filters are checked. For secondary system initialization, parameters are requested from the primary system.

*Subroutine INIT_P2*

This is the phase two initialization for SYSMON. It initializes those system components that must be reconfigured when a reinitialization occurs. INIT_P2 is called once at system startup, and again for the secondary system when a primary CPU failure is detected and the secondary has to reinitialize itself as primary. NAVSPASUR communications, data processing, and target selection procedures are started.

*Subroutine INTERR (INUM)*

This routine allows linear handling of internal error messages. It notifies the SYSMON procedure by sending it a packet containing INUM, the message number. The message number is translated into text by subroutine FMTMSG and printed by subroutine PRNTMSG.

*Subroutine MODIFY*

This routine allows an operator to modify system parameters (e.g., antenna, target selection, calibration, and hardware.) It displays the modify commands and what information can be modified by each command. When a command (single character) is typed into the console, another menu is displayed, showing the individual items that can be modified for that system parameter. Only at the individual item level can modifications be made.

*Subroutine NL_DISPLAY*

When notching is enabled, this routine outputs to the primary display the full-Doppler notch list, and to the secondary display the half-Doppler and quarter-Doppler notch lists.

*Subroutine PMTR_XFER (N)*

This routine receives interprocessor communication commands and messages and dispatches control to routines to comply with the commands. The messages are of two types: send and request. A send message means that information has been sent to this module. A request message means that information must be sent from this module to SYSMON on the other system. Argument N is the interprocessor communication sub-buffer number.

*Subroutine PRNTMSG (ERR)*

This module is the system error-handling routine. Other tasks in the system alert SYSMON of an error by sending a packet containing an error message number. When an error has been reported, an appropriate message and time stamp are output to the console CRT and hardcopy terminal. PRNTMSG calls subroutine FMTMSG to format the message for display. Argument ERR is the number of the error to output to the terminals.

For a fatal error, a call to ALARM is made to notify the operator and calls to ABORT_TASK are made to cancel all necessary procedures to shut down the system. For a warning error, only ALARM is called. A second entry is provided for calls from the asynchronous trap (AST) routine, which has already formatted the error message.

*Subroutine PRNT_DLY_MSG (BUFNUM)*

This module is used to print data delay errors. Argument BUFNUM is the number of the ICC buffer that is used.

*Subroutine REC_1*

This module is the dispatch routine for level 1 messages received asynchronously by SYSMON. These are send/receive packets 15 words in length that are received in SYSMON common area SMCOM to return responses and report error conditions or status changes. The first two words of each packet are used for the source and destination process numbers. The third word of the 15-word message indicates the type of message, and the additional words contain the message data itself. (See Appendix G in Volume 1 for further explanation of send/receive packets and message levels.)

*Subroutine REQ_OPTEST*

This routine displays the operational test (OPTEST) menu and carries out the operator's selection of which operational tests to run. The operational tests are the RF calibration test (RFCAL), the digital filter test (OPDFT), the alert sensitivity test (OPALRT), and the system signal confirmation test (OPSYS). There is an automatic mode keyed to the DSPCOM variables MODPC and MODRF. If MODPC is non-zero, operational test OPSYS is automatically run; if MODRF is non-zero, an automatic RFCAL is run at boot time and at scheduled time intervals.

*Subroutine SCLOCK*

This routine prompts the system operator for a command to update the Hewlett-Packard 59309A digital clock on the primary or secondary systems, or on both. It waits for the command to be received or a time-out to occur. If a command is received, a request to update the clock is sent to procedure CLOCK. If a time-out occurs, the dynamic display is reactivated.

*Subroutine SCRASH*

This module is called whenever a system crash occurs. A message is output to the operator notifying that a system has failed, and appropriate system variables are set to reflect this. If the CPU that has failed is the primary system CPU, then the secondary system CPU will reinitialize itself as the primary system.

*Subroutine SCSHUT*

This routine shuts down the DSPR system in an orderly manner. The operator can choose to shut down the primary system, the secondary system, or both systems. If both systems are to be shut down, the secondary system is shut down first by sending an ICC buffer to the secondary system with a shutdown message. When the primary system is shut down, subroutine ABORT_TASK is called for all tasks except SYSMON. If only the primary system is to be shut down, ICC is also not aborted.

*Subroutine SEC_DISP*

SEC_DISP is the secondary SYSMON display and activity monitor. It displays the text "SEC" and the date and time on the secondary system CRT. When notching is enabled, it also displays the half- and quarter-Doppler notch lists. The time is updated every 10 seconds. While the secondary system is active, the number of targets processed per digital filter is shipped via ICC to the primary system once every minute. If no targets are processed in 3 minutes while the system is active, an error condition is declared. The routine is exited when reinitialization is set or when a shutdown message from the primary system is received.

*Subroutine SMDTXT (LX1, LX2)*

This module inserts two characters (LX1 and LX2) into a buffer and sends a message to SYSMON.

*Subroutine SYSACTIVE*

This routine processes the operator's GO command. This allows the system to accept information from the array processor lists and start data acquisition. The routine also calls REQ_OPTEST to perform an automatic reference calibration.

*Subroutine SYSIDLE*

This routine processes the operator's IDLE command. This allows present data to pass through the system, but bars acquisition of any new data.

## 2.2 Macro Modules

Routines described below comprise the system monitor subroutines written in the VAX Macro assembly language. Routines RCVAST and UNSOL enable system asynchronous traps, which allow asynchronous processing to take place within the usually synchronous Fortran processing system. The ASTs enabled are a receive-data AST and an unsolicited-character-input AST.

*Subroutine DEL_MAILBOX*

DEL_MAILBOX marks mailboxes for deletion.

*Subroutine FMTMSG*

This routine receives a message number from subroutine PRNTMSG and from that number generates the appropriate informational/error message. Associated with every message number is text that describes the condition. Also included in the text is the severity of the error: fatal, warning, or log. The data structure for the informational message is built in a message buffer ERMSG in common area SMCOM, and can consequently be output to the console terminal and/or hardcopy terminal. The time and date are also put into the data structure so that the message may be properly stamped with the system time.

*Subroutine GETCHAR*

GETCHAR processes a character that the operator types on the console. This routine is activated by UNSOL, which accepts unsolicited input.

*Subroutine INIT_MAILBOX*

This routine initializes mailboxes for SYSMON to use in sending and receiving messages from other processes.

*Subroutine INIT_TASK*

This routine creates a detached process and starts a task running in that process. The two formal arguments are IMAGE, the description of the task to start, and PID_ADD, the address in which the process identification is stored.

*Subroutine INIT_TIMER*

INIT_TIMER initializes timer values for SYSMON to use in setting time-out or wait values.

*Subroutine RCVAST*

This routine handles all receive-data ASTs for the system monitor SYSMON. It dispatches to subroutine REC_1 for simple (level 1) messages. It dispatches to subroutine REC_2 for longer (level 2) messages associated with ICC. For level 1 error reporting on the secondary system, an ICC buffer is allocated and formatted with the error message, and then sent to the primary system via the ICC link. If the error on the secondary system is fatal, an attempt is made to write the error on the secondary system CRT and the hardcopy terminal via a call to the ASTERR entry point in subroutine PRNTMSG.

*Subroutine REC_2*

This routine handles level 2 messages to the system monitor routine SYSMON. All input is in the form of an ICC buffer coded for the appropriate function.

*Subroutine RUN_RFCAL*

RUN_RFCAL sets event flag RFCFLG to "true" to request an automatic reference calibration operational test.

*Subroutine UNSOL*

UNSOL accepts unsolicited input in the form of a character typed by the operator.

## 2.3 Hardware Components

### *2.3.1 Central Controller*

The hardware of the Central Controller subsystem consists of a Digital Equipment Corporation VAX 4000 Model 200 (VAX 4200) with attached peripherals and communications components. The system monitor and most of the applications software execute in the VAX 4200. The exception is the target detection software, which executes in the MAP 4000 array processor.

The VAX 4200 runs the operating system VMS version 5.4-3 and uses the Qbus as its communications bus. It has 32 MB of central memory; its processing power is 4.8 million instructions per second (MIPS). It operates at a clock rate of 114 MHz and has a 64 bit wide data path. The CPU is installed in a BA430 10-slot rack mounted chassis with a B400X 11-slot expansion chassis. The bus for peripherals is a 22 bit Qbus with a maximum I/O bandwidth of 2.4 MB for reading and 3.3 MB for writing.

### *2.3.2 Operator's Console*

The operator's console serves as the interface between the operators and the DSPR system and allows them to control the system and obtain system status information. It consists of a VT420 video terminal connected to the central controller through a DZQ11 serial line interface and an LA120 printing terminal connected to the console port on the CPU. Only the devices on the primary system are active. The operators enter commands on the VT420 through a menu-controlled software interface. During normal operation, dynamic status information is displayed. The LA120 is used to generate a hardcopy log of system performance. All messages from the system and results of operational tests are displayed both on the VT420 and the LA120. On the secondary system, the LA120 is disabled, and the VT420 displays the text "SEC."

### *2.3.3 DZQ11 Serial Line Interface*

The DZQ11 is a four-line asynchronous communications controller compatible with the RS-232-C serial line interface standard. It serves as a character-buffered communication interface designed to assemble or disassemble the serial information required by the LA120 line printer. Data rates are selectable from standard rates between 40 and 9600 baud; in the DSPR system, it operates at 9600 baud. (For a complete description, see *DZQ11 Asynchronous Multiplexer User's Guide*.)

### 2.3.4 *VT420 Video Terminal*

The VT420 is a high-performance general purpose interactive video display terminal. It uses a 14-inch monochrome screen and a DEC LK401 keyboard. Feature settings are stored in nonvolatile memory and include a screen display of 24 to 48 lines of text in 80 or 132 columns. Transmit and receive speeds can be set from 300 to 38,400 baud; in the DSPR system, it operates at 9600 baud. (See *Installing and Using the VT420 Video Terminal.*)

### 2.3.5 *LA120 Hardcopy Terminal*

The LA120 Hardcopy Terminal (DECwriter III) is a pedestal-mounted, interactive, high-speed smart printing terminal. It has a 1 K character buffer and can print bidirectionally at 180 characters per second. The LA120 operates at 9600 baud in full-duplex mode using a RS-232-C interface connected to line zero of the DZQ11. (See *LA120 User Guide.*)

# 3. TARGET DETECTION AND SELECTION

This subsystem performs the initial detection of targets that enter the NAVSPASUR fence and compiles target lists that are used for target selection. Target detection and selection is divided into two parts: the target detection procedure, which runs in the MAP 4000 array processor (AP), and the target selection procedure (TRGSEL), which runs in the VAX 4200 central processor.

## 3.1 Fortran Modules

*Program TRGSEL*

This program is responsible for initializing the MAP 4000 array processor and starting its generation of target lists. The target lists are 21-word arrays that contain 19 targets, plus two words reserved for other information. TRGSEL selects from the list targets on which to collect data and maintains elimination and notch lists to prevent redundant gathering of data. In addition, the data collection tasks ADC1, ADC2, and ADC3 are started by this program.

*Subroutine LOADMAP (ICMD, APLUN, STATUS)*

This routine loads code into the MAP 4000 to initialize the array processor for operation in the primary or secondary system. Argument ICMD indicates primary or secondary system. APLUN is the error code for initialization of the MAP. STATUS is set to 1 for successful completion of the LOADMAP routine or to -1 for failure.

*Subroutine RECFREQ (BUFNUM)*

This module is responsible for receiving data from the target selection procedure on the primary system. It receives the frequency (which is actually an FFT bin number) of a target on which the primary system is collecting data. It then enters this information into the half- and quarter-Doppler target elimination lists. Before it can enter the data into the lists, however, it must convert from the full-Doppler bin number to the corresponding half- and quarter-Doppler bin numbers. The relationship is such that a signal with bin number $n$ in the full-Doppler region would have bin number of $4n$ in the half-Doppler and bin number of $8n$ in the quarter-Doppler region. Argument BUFNUM is the ICC buffer number.

### 3.2 Macro Modules

*Subroutine EDT_TRGLIST*

This routine compares the entries in the AP list to elements on the notch list. If a bin is on the AP list (from array NLTMP) and on the notch list (within tolerance), then that bin is not copied to TRGSEL array LIST. If the AP bin (from NLTMP) is not on the notch list, then it is copied to array LIST. Any space not used in the array is zeroed out. The Fortran call is CALL EDT_TRGLIST (NOTCH_ADDR). Argument NOTCH_ADDR is the address of the notch list which was input.

*Subroutine ERROR_CHECK*

This routine checks for an error condition after executing a system routine. If there is an error condition, LIB$SIGNAL handles the error. An error condition in a system routine is indicated by the low bit clear in register R0.

*Subroutine GET_TRGLIST*

This module starts the array processor generating 21-word lists of targets, and copies the lists into common area TSCOM for use by target select. It also performs some calibration processing by counting the number of array processor lists and the number of occurrences of the calibrated signal on those lists, whenever the calibrator is on. It calls subroutines SELECT_NLIST, EDT_NLIST, UPDATE_NLIST, and MAINT_NLIST to process notch lists.

*Subroutine MAINT_IHALF*

This module maintains the half-Doppler elimination list in common area TSCOM. The list is made up of targets previously selected from the full-Doppler region. The list also contains a number called the list-generated count. Each time a target list is generated for the half-Doppler region, the list-generated count is decremented if the count is non-zero. Since lists are generated for the half-Doppler region at a rate of approximately nine per second, this table maintains the list of targets chosen over the past few seconds. Targets selected for the half-Doppler region are also added to this list.

*Subroutine MAINT_IQUART*

This module maintains the quarter-Doppler elimination list in common area TSCOM. The list is made up of targets previously selected from the full-Doppler and half-Doppler regions. The list also contains a number called the list-generated count. Each time a target list is generated for the quarter-Doppler region, the list-generated count is decremented if the count is non-zero. Since lists are generated for the quarter-Doppler region at a rate of approximately two per second, this table maintains the list of targets chosen over the past few seconds. Targets selected for the quarter-Doppler region are also added to this list.

*Subroutine MAINT_NLIST*

This routine shuffles the notch list to move valid notch bin numbers to the top of the list. It also counts the valid bin numbers. The notch list is two-dimensional, with the first dimension as the bin number and the second as the count. If the count for a bin is zero, then that slot is used in the shuffle. The Fortran call is CALL MAINT_NLIST (NOTCH_ADDR). Argument NOTCH_ADDR is the address of the notch list that is input.

*Subroutine MYWAIT*

This routine sets up a timer and waits for a given number of seconds. It is identical to the utility routine WAIT, except that the name has been changed because the word WAIT is reserved for use by the MAP code. The Fortran call is CALL MYWAIT (%val(EFN, DAYTIM), where EFN is the event flag number and DAYTIM is the address of the quad-word containing the time value.

*Subroutine RCVMSG*

RCVMSG receives a send packet. Its type (PRI_ELIM) indicates that it contains an elimination frequency from the primary system. It calls routine RECFREQ to handle the message, then it reposts a read.

*Subroutine SELECT_ADC*

This routine places the bin number (frequency) of the target selected into global common area DSPCOM and requests the appropriate data collection task to run. The frequency of the target is sent to the secondary system, if this system is primary and if the other system is functioning. The Fortran call is CALL SELECT_ADC (BINNUM, ADCNUM). BINNUM is the input bin number; ADCNUM is the number of the digital filter that is free, which indicates which ADC task to use.

*Subroutine SELECT_NLIST*

This routine is called if notching is enabled. The first part of the routine determines which notch list array to use (NOTCH1, NOTCH2, or NOTCH3, associated with full-, half-, or quarter-Doppler targets). Once the correct notch list is determined, the routine tests it to see if it is empty. If the notch list is empty, the routine simply sets the low bit in register R0 to discontinue further execution of notching routines and returns control to GET_TRGLIST. If the notch list is not empty, then a check is done on the array processor list APLIST. If the APLIST is not empty, it is copied to a temporary notching array (NLTMP), the carry bit is cleared to indicate to routine GET_TRGLIST that the rest of the notching routines are to be executed, and the routine returns control to GET_TRGLIST. If the APLIST is empty, the notch list count-down values are decremented, the notch list is compacted, the low bit of R0 is set to discontinue further execution of notching routines, and control is returned to GET_TRGLIST. Upon each return to the calling routine GET_TRGLIST, the address of the correct notch list is passed to GET_TRGLIST in the argument NOTCH_ADDR for further use. The Fortran call is CALL SELECT_NLIST (NOTCH_ADDR).

*Subroutine TARGET_CANCEL*

This routine cancels collections if the target is not on the array processor list more than 'n' times, where 'n' is set in global common area TRGCOM.

*Subroutine TARGET_CHECK*

This routine makes sure that the Doppler region of the list from the array processor agrees with the system setup and checks for idle data collection tasks. The status of each digital filter is checked to make sure it is operational before it is considered for allocation. If no data collection tasks are idle (i.e., all digital filters are already in use), control is returned to the calling routine TRGSEL. If a digital filter is available, target selection is initiated using that digital filter. When a data collection task is started and a digital filter is put in use, the collection task variable COLn (where 'n' is the digital filter number) is

set to 1 to indicate a status of "in use." It accesses, via an AST, subroutines RCVMSG and RECFREQ to receive frequencies for the elimination lists; it calls subroutines TARGET_CANCEL, MAINT_IHALF, MAINT_IQUART, TARGET_ELIM, and SELECT_ADC to process elimination lists.

*Subroutine TARGET_ELIM*

This module is designed to eliminate redundant gathering of data by checking to see if the current target candidate for a particular Doppler region is on that region's target elimination list. The list for the primary system (full-Doppler) region is three words in length, with entries of the bin number of data collected. The secondary system (half-Doppler and quarter-Doppler) regions have elimination lists that are twice as long as the variable LSTLEN. The first word corresponds to the bin number of a signal and the second to a time count. These lists contain the bin number of a signal on which the primary system is collecting data, as well as the bin number on which this region is collecting data. The time count is a method for ensuring that redundant targets are eliminated if the system has collected data on that bin number within the past few seconds. Each time an array processor list is received by the secondary system, the time count field for that region's elimination list is then decremented. There are approximately nine lists that are generated for the half-Doppler region every second and two lists for the quarter-Doppler region. If the time count of a target is not zero, then a comparison must be made to protect against redundant gathering of data on similar signals. The Fortran call is CALL TARGET_ELIM (INDEX). The argument INDEX contains the input index into the target list and also receives the output index to the target list.

*Subroutine UPDATE_NLIST*

This routine updates the target notch list. If an element on the notch list, including its tolerance, does not appear anywhere in the array NLTMP, which is a temporary copy of the array processor list APLIST, then that notch list counter is decremented. If the element on the notch list, including its tolerance, does appear at least once in the NLTMP array, then the notch list counter is initialized to 20. Each element of the notch list is checked against each element of the array NLTMP. The Fortran call is CALL UPDATE_NLIST (NOTCH_ADDR), where NOTCH_ADDR is the input address of the notch list to be used.

### 3.3 MAP 4000 Modules

These subroutines execute in the MAP 4000 array processor. They were compiled with the MAP 4000 Fortran compiler, version 2.1.2. Their file type is .FTN.

*Program PMAIN*

PMAIN is the main MAP routine for the primary system. It initializes the direct input/output (DIO) board to begin double buffer transfers and calls FULL_SP to process each full-Doppler alert data buffer.

*Program SMAIN*

SMAIN is the main MAP routine for the secondary system. It initializes the direct input/output board to begin double buffer transfers, then calls HALF_SP to process each half-Doppler alert data buffer and QUART_SP to process each quarter-Doppler alert data buffer.

*Subroutine BHF*

This subroutine generates a 4096-point Blackman-Harris window for use by the full-Doppler processing.

*Subroutine BHS*

This subroutine generates a 16k-point Blackman-Harris window for use by the half-Doppler processing.

*Subroutine FULL_SP*

FULL_SP performs the signal processing of the full-Doppler alert data. It converts the data to floating point form, windows the data using the window generated by routine BHF, and performs an FFT on the data. It also detects targets and generates target lists. It is called by program PMAIN.

*Subroutine HALF_SP*

This module performs the half-Doppler processing on the secondary system. It converts the data to floating point form, windows the data using the window generated by routine BHS, and performs an FFT on the data. It also detects targets and generates target lists. It is called by program SMAIN.

*Subroutine INSERT*

This routine inserts a new target onto the target list. Since the list is in descending order by magnitude, this routine inserts the new target in its appropriate place of the LIST array.

*Subroutine PRI_NOTCH*

This routine contains code for the full-Doppler feedthrough notch filter using a Chebyshev bandstop digital filter. The calculations for each section are performed by subroutine FSECT, whose code is included in PRI_NOTCH.

*Subroutine QTR_SP*

This module performs the quarter-Doppler processing on the secondary system. It performs four-point discrete Fourier transforms (DFTs) using successive half-Doppler FFT points, detects targets and generates target lists. It is called by program SMAIN.

*Subroutine SEC_NOTCH*

This routine codes the half-Doppler feedthrough notch filter using a Chebyshev bandstop digital filter. The calculations for each section are performed by subroutine HSECT, whose code is included in SEC_NOTCH.

### 3.4 Hardware Components

Hardware used by the Target Detection and Selection subsystem consists of the alert antenna, the RF subsystem, the array processor, the array processor input/output interface card, and the main data bus. The alert antennas are a subset of the receiver antennas, whose outputs are combined electrically to

provide a usable signal before the interferometer antennas do. The RF subsystem converts the analog outputs from the antennas to digital form.

### *3.4.1 CSPI MAP 4000*

The MAP 4000 is an array processor designed to operate as an attached processor for DEC Qbus computers. Its consists of four cards which are designed to plug directly into the Qbus: a host interface card, the CPU card, a main memory card, and a DIO 16 bit parallel interface card for external devices. A shared memory region is used to transfer data to main memory of the CPU. It is capable of operating at 40 megaflops (million floating-point operations per second). MAP 4000 capabilities and functions are described in the CSPI manuals listed in References.

### *3.4.2 APIO Interface Card*

The Array Processor Input/Output (APIO) card is a custom device designed to select alert antenna data from the main data bus and transfer it into the MAP 4000. It extracts alert channel data from the continuous stream of antenna data passing through the Data Distribution and Test Card and transfers the data to the MAP through the MAP's DIO board. It also reformats the antenna data to accommodate the MAP input format requirements.

### *3.4.3 Main Data Bus*

The Main Data Bus is a custom bus designed to handle the digitized outputs of the DSPR RF channels. It is time multiplexed with 16 time slots operating at a 75 kHz rate. It has 16 parallel data lines and contains several control signals.

## 4. INTERFEROMETER DATA COLLECTION

The Interferometer Data Collection subsystem takes the targets selected by the target selection procedure and tunes the digital filters to gather data. Accumulated data are provided to the central controller for formatting and transmission to the NAVSPASUR Processing and Operations Center at the NSSC in Dahlgren. Procedure ADC controls this task. Hardware used by the subsystem includes the six digital filters (three each for the primary and secondary systems), the Data Distribution and Test Card, and three DRV1W interfaces, one for each digital filter.

### 4.1 Fortran Modules

*Program ADC*

The antenna data collection (ADC) process controls the collections of data from the interferometer antennas. On each system (primary and secondary), the process is made up of three programs—ADC1, ADC2, and ADC3—each associated with one of that system's three digital filters. Once a target is detected and selected, an ADC task is requested to run. The task first gets the frequency to which the digital filter should be tuned from global common area DSPCOM, where it was placed by the target select program. The digital filter is then set to the correct frequency and bandwidth, and a reset command is given to signal the digital filter to start collecting data. The collected data are time stamped. When the collection is finished, the data processing task is notified with the setting of an event flag. The array to set up the digital filter is as follows:

| | |
|---|---|
| 1st word | 1st word of DRV1W set bandwidth command |
| 2nd word | 2nd word of DRV1W set bandwidth command |
| 3rd word | 1st word of DRV1W set frequency command |
| 4th word | 2nd word of DRV1W set frequency command |
| 5th word | DRV1W reset command |
| 6th word | DRV1W no operation command |

## 4.2 Macro Modules

*Subroutine GET_TIME*

GET_TIME reads the current time from the system time, RFTIME, and stores it in data buffer ITIM. The Fortran call is CALL GET_TIME (RFTIME, ITIM).

*Subroutine MAP_IO_PAGE*

This routine uses physical pages to map to the Qbus I/O page.

## 4.3 Hardware Components

### *4.3.1 Digital Filters*

The digital filters (DFs), which were designed and built by NRL, measure the signal on each of the interferometer antennas by performing a single-frequency discrete Fourier transform on the digitized antenna data. Each DFT generates a narrow bandpass filter at the selected frequency. The width of the filter is dependent on the number of points used in the DFT and the type of window used. The DFs use a Blackman-Harris four-point-minimum sidelobe window. The full-, half-, and quarter regions use 4096, 16384, and 65536 points respectively, which translate into bandwidths of 36.6 Hz, 9.1 Hz, and 2.28 Hz.

The DFs receive time division multiplexed digitized antenna data at 75 kHz. They also receive 75 kHz and 1.2 MHz reference frequencies from the RF subsystem via a Data Distribution and Test Card, which attached to the central controller. Only the first 14 antennas on the data bus are processed. Data are processed one point at a time using a 3:1 overlap. This means that the DFs are performing 42 simultaneous DFTs. When a target is selected and the appropriate antenna data collection task is activated, the ADC task sends frequency, bandwidth, and reset commands to the DF. A 60 MHz signal is provided by the RF subsystem via the 60 MHz distribution amplifier.

Each digital filter is composed of a Digital Filter Card 1 and a Digital Filter Card 2.

Digital Filter Card 1: Digital Filter Card 1 accepts frequency and bandwidth commands from a DRV1W and 60 MHz from the distribution amplifier. It also provides control, timing, and window coefficients to Digital Filter Card 2 for processing the digitized antenna data.

Digital Filter Card 2: Digital Filter Card 2 accepts the digitized antenna data along with 75 kHz and 1.2 MHz from the Data Distribution and Test Card. Digital Filter Card 2 accepts control, timing, and coefficient signals from Digital Filter Card 1 and performs the DFTs. As each DFT is completed, the complex signals are transmitted to the VAX CPU through the DRV1W interface.

#### *4.3.2 Data Distribution and Test Card (DD&TC)*

The DD&TC acts as a buffer between the main data bus and the digital filters. The DD&TC fans the data and timing signals to the individual DFs on each side of the DSPR. The card was designed by NRL and custom built for the DSPR system.

In addition, the DD&TC supports off-line diagnostics by providing digital test signals for the Interferometer Data Collection and the Target Detection and Selection subsystems. The three test signals generated are a sine wave at a variable frequency, a ramp, and a constant level.

#### *4.3.3 DRV1W Interface*

The DRV1W is a general purpose 16 bit parallel direct memory access (DMA) interface for Qbus systems. It has 22 bit addressing capability and permits data transfers at rates up to 250,000 words per second. There are four DRV1Ws on each VAX 4200: three for the DFs and one for the Utility Bus interface. The DRV1W has several modes of generating interrupts for input conditions. The DSPR system uses external control lines to generate interrupts. Each device (card) on the utility bus coordinates its I/O with the DRV1W via these control lines. (See *DRV11-WA General Purpose DMA Interface User's Guide*.)

## 5. DATA PROCESSING

This subsystem formats the data collected by the Interferometer Data Collection subsystem for transmission to the NSSC. Procedure DP controls this task. Hardware components are the same as those used by the System Monitor and Control subsystem.

### 5.1 Fortran Modules

*Program DP*

DP is the main procedure for data processing. The system monitor procedure initializes and starts up DP, but DP waits until event flag DCOLEF is set by an ADC task before starting any data processing. The interferometer data processing task then selects data collected by a digital filter, processes the data, and sends the results in a transport control buffer to the data line communication (DLC) procedure for transmission to the NSSC. Data from the secondary system are processed in the same way as the primary system data, except that the results are sent to the primary interprocessor communications (ICC) procedure in an ICC buffer, where program GETSEC puts the data into a transport control buffer. DP directs this task by calling the appropriate subroutines. When processing is complete, the appropriate collection task status variable (COL1, 2, or 3) is set to idle to indicate to the target selection procedure that processing has been completed on one target.

*Program GETSEC*

GETSEC is a separate task which creates a secondary system transport control buffer (TCB) from half- or quarter-Doppler observation data received from the ICC procedure on the secondary system. The data are received in one to seven ICC buffers. The 128 byte buffers, which have been byte-packed to use every byte in each buffer (except the last buffer), are received one at a time, in sequential order from ICC. GETSEC assigns the data in the ICC buffers to data transmission packets, which form the transport control buffer TCB4, until the total number of buffers for that observation has been received. GETSEC

then sets event flag DLCEF to notify program DLC that TCB4 is ready for transmission to the Processing and Operations Center at the NSSC.

*Subroutine FILL_PRI (TCBBUF)*

The fill primary routine fills the appropriate TCB, which was assigned to argument TCBBUF in program DP, with formatted data from the primary system for transmission via DLC to the NSSC. It is composed of from one to six 128 byte data transmission packets (DTPs). The first packet contains a packet header and other general information about the data processing task, such as time at epoch, duration of observation, Doppler region and Doppler frequency, a signal strength deviation block, and up to seven full time-lines of antenna phase data. If there are unfilled bytes after entering the last full phase block in a packet, then those last bytes are assigned the signal strength and some phases from the phase block until the remaining bytes are filled. The partially filled phase block is continued in the next packet. Subsequent 128 byte DTPs (up to five more) have a packet header only, and are filled with up to nine full time-lines of antenna phase data, plus the remaining bytes of partial time-lines at beginning and end. FILL_PRI continues to fill DTPs until all of the time-lines from the current data processing task (up to 55 time-lines) have been stored in the TCB buffers. At this time, the total number of packets used is reported in TCBFLG(ITRG) and the global flag DLCEF, which activates the DLC task, is set.

*Subroutine FILL_SEC*

The fill secondary routine requests a 128 byte ICC buffer, fills it with formatted data from the secondary system, and sends it to the primary system via ICC and GETSEC for transmission via DLC to the NSSC. From one to seven 128 byte ICC buffers are sent per observation.

The first buffer contains an ICC buffer header, a packet header, a header control block with general information about the data processing task, such as time at epoch, duration of observation, Doppler region and Doppler frequency, a signal strength deviation block, and up to seven time-lines of antenna phase data. Subsequent 128 byte ICC buffers (up to six more) have an ICC buffer header and packet header only, and then are filled with up to nine time-lines of antenna phase data. FILL_SEC continues to fill and send ICC buffers until all of the time-lines from the current data processing task (up to 55 time-lines) have been transmitted. If there is no room at the end of an ICC buffer for all 13 bytes of a whole phase block, then as many bytes as are left are packed with part of a phase block. The rest of that phase block is inserted in front of the phase blocks in the next ICC buffer.

Since the telephone data lines are connected only to the primary system, ICC buffers are used to send the secondary data to the primary system, where program GETSEC formats them into TCBs, and notifies program DLC that it is ready to send them to the NSSC.

*Subroutine FILTER*

This routine filters out target frames that meet the feedthrough parameters stored in global common area DSPCOM (frame duration, Doppler range, and signal strength). Communication with the main routines as to target frame validity is through flag XMIT in DP common area DPCOM.

*Subroutine GET_DATA*

This module moves data from the target buffer to the data processing procedure's internal buffer. If spare channel substitution is required, it performs the substitution.

*Subroutine GET_WORK*

This module is responsible for getting work for the main data processing routines. It waits for event flag DCOLEF to be set by an antenna data collection task. This indicates that a digital filter has finished collecting data and that the data are ready to be processed. The data collection task status variables (COL$n$, where $n = 1, 2,$ or 3 ) are checked for a status of "finished." The digital filter which collected the calibration data is identified and is marked by setting the variable ITRG.

*Subroutine INIT_DP*

This module initializes the user database, common area DPCOM, with values of variables from the system databases, global common areas DSPCOM and TRGCOM. (It "copies" the values one for one). The system database is locked while values are copied into the user common to prevent any other module from gaining access to the database. The databases are then left in an unlocked state.

*Subroutine NOTCH*

NOTCH selects the proper notch list (full-, half-, or quarter-Doppler), and increments the counter NLARGn. The frequency IFRQ (FFT bin number) and the notch list count-down value NLCNT are then entered on the notch list NOTCHn.

*Subroutine PHASE*

The DP phase calculation subroutine separates the real and imaginary components of the complex data value and uses them as parameters in a user-written Fortran integer function named PHASE_ANGLE. PHASE_ANGLE performs an arctangent calculation on these values and returns the phase angle in eight-bit representation. This phase value is stored in the byte array BBUF. (PHASE uses the value of DURATN to transmit the complex antenna data to PHASE_ANGLE.)

*Integer Function PHASE_ANGLE (Y, X)*

Function PHASE_ANGLE determines the phase angle in eight-bit representation by performing an arctangent calculation upon the imaginary and real components of each observation in the antenna data collection. The argument Y is the imaginary component and the argument X is the real component of an observation received in an antenna data collection.

*Subroutine RECORD_DATA*

RECORD_DATA is a debug module which records raw digital filter data from an observation into an unformatted file. This data file also contains calculated values that are related to the observation. A configuration file contains values in effect at the receiver station at the time of the observation. The names of these two files reveal the name of the station from which the data were taken and the date on which they were taken. For example, the configuration file from the receiver at Silver Lake, Mississippi, that contained the first data extracted on June 10, 1986, would be named M860610.C1. The data from that observation, with the calculated header values, would be stored in a related file named M860610.D1.

*Subroutine RELEASE_DF*

RELEASE_DF resets the collection status variable and removes a frequency from the elimination list for non-calibration data.

*Subroutine SIGNAL_DBM*

The DP log calculation subroutine uses the values of PRFSTR (average signal strength for each frame of an observation) and CHNSTR (average signal strength for each channel in a frame) to determine signal strengths in decibels above 1 milliwatt (dBm) and to calculate the signal strength deviations.

*Subroutine SS_PROFILE*

SS_PROFILE calculates the individual channel strength and uses an "olympic" average to form a signal strength profile for the 55 time-lines.

### 5.2 Macro Modules

*Subroutine DV_ZERO*

DV_ZERO is an asynchronous error handling routine.

*Subroutine SETFPX*

SETFPX moves the address of DV_ZERO to the routine header frame to allow it to handle exceptions.

## 6. DATA LINE COMMUNICATIONS

This subsystem is responsible for coordinating the flow of data between the central processors in the receiver stations and the Processing and Operations Center at the NSSC. The data line between each station and the NSSC is a leased C1 conditioned line that operates at 9600 baud in asynchronous mode. The data line communications (DLC) procedure completes the formatting of data frames and passes them to a DSV11 synchronous line interface for transmission.

### 6.1 Fortran Modules

*Program DLC*

Program DLC sends data buffers to the NSSC. Data have been collected by one of the ADC programs (discussed under Interferometer Data Collection) and have been assembled by program DP, for primary data, or GETSEC, for secondary data (both discussed under Data Processing). It also sends and receives keep-alive messages to maintain connection with the NSSC.

*Subroutine INIT_DSV (IERR)*

INIT_DSV initializes a DSV11. The DSVs are used to communicate between the two CPUs and to send information to the modem and down the communications line to the NSSC. Argument IERR is the return value of an error.

*Subroutine RE_TRANS*

RE_TRANS retransmits frames of data to the NSSC.

*Subroutine SEND_FRAME (INBUF)*

SEND_FRAME sends a frame of data, INBUF, to the NSSC.

## 6.2 Macro Modules

*Subroutine DL_READ*

This module is an AST routine that is entered when a message is received from the NSSC acknowledging transmission of data from a receiver station. It checks the received and sent counters to see if the transmission to the NSSC was successful. If so, it releases the data buffers that had been sent to the NSSC. If not, it calls RE_TRANS to retransmit the data.

*Subroutine INIT_CIR*

This module initializes a synchronous data line control (SDLC) circuit as a secondary (slave) node.

*Subroutine KA_TMO*

KA_TMO tests to see if the data line to the NSSC is up. If so, it enables sending a packet over the line by setting the KA_SEND flag and informing program DLC of the event by setting event flag DLCEF. If the data line is not up, then it clears event flag KA_TMO_EF and resets the timer for the next check on data line availability.

*Subroutine READ_TMO*

This module, if the data line to the NSSC is up, reinitializes the data line and sends a recovery message to SYSMON. If the data line is down, then it marks it down and checks for data line failure. If the failure is not a data line failure, then it checks for DSV failure. If the failure is not in the DSV, then it checks for modem failure and sends a data line failure message to SYSMON.

*Subroutine RR_TMO*

RR_TMO sends receiver-ready messages to the NSSC every 10 seconds to keep the data line alive.

## 6.3 Hardware Components

### *6.3.1 DSV11 Synchronous Communications Controller*

The DSV11 is a dual-channel synchronous communications controller for Qbus systems. It supports the DDCMP, HDLC, and SDLC protocols at speeds up to 256 kilobytes per second (kb/s) for one-line-only operation and speeds up to 64 kb/s per line for two-line operation. In the DSPR system, only one line is used at 9600 baud. (See *DSV11 Communications Option Technical Description* and *VAX Wide Area Network Device Drivers, Installation Guide, Specifications, and Programmer's Guide*.)

### *6.3.2 Codex V.3225 Modem*

The Codex V.3225 modem is the interface between the DSPR system and the phone lines to NAVSPASUR Processing and Operations Center. The modem operates in full duplex at 9600 baud in

trellis-coded mode as per recommendation V.32 of the Comité Consultatif Internationale de Télégraphique et Téléphonique (CCITT). (See *V.3224/V.3225 Installation and Operation.*)

## 7. INTERPROCESSOR COMMUNICATIONS

The two CPUs communicate over a thinwire Ethernet link connecting the two systems. The interprocessor communications (ICC) procedure routes message from subsystems on one CPU to subsystems on the other.

### 7.1 Fortran Modules

*Program ICC*

This process controls the communications between the two VAX 4200s through the Ethernet controller. First, a channel is assigned to the controller and it is initialized. A read QIO with a 30-second timeout is posted. If the read times out, the process assumes that the other system is not operating as a DSPR system and a mode of 1 (primary system) is returned to SYSMON. If the read completes, it means the other system is operating and a mode of 2 (secondary system) is sent to SYSMON. ICC then attempts to initialize the circuit in accordance with the mode. After initialization, a keep-alive timer of 2 seconds is set with an AST routine of KA_TMO and a read is posted along with a 10 second timer. If the read completes, AST routine COM_READ executes and resets the read QIO and the timer. If the timer times out, it indicates that communication with the other system has failed. In that case, AST routine READ_TMO executes and notifies SYSMON of the failure. A write attention is associated with the ICC mailbox, with AST routine READ set to execute when a message is received by the mailbox. READ checks the message and calls SEND_FRAME to queue it to the Ethernet driver for transmission to the other system. When the keep-alive timer expires, the system status is placed in a message buffer and sent to the other system.

*Subroutine INIT_ETHNET (IERR)*

INIT_ETHNET initializes the Ethernet controller, which is used to communicate between the two CPUs. It posts a read QIO to the Ethernet logical unit number. If the read completes, it means that the other system is operating and this system should initialize as a secondary system. Otherwise, it should initialize as a primary system. Argument IERR is set to -1 if there is an initialization failure.

*Subroutine RESTART*

RESTART restarts the Ethernet controller when a communication failure occurs.

*Subroutine SEND_FRAME (INBUF)*

SEND_FRAME moves the data from an ICC buffer to a transmit buffer (INBUF) and sends the buffer to the other system with a write QIO to the Ethernet driver.

*Subroutine WRTMSG*

WRTMSG is called when a message from the other system has been received. It allocates an ICC buffer and fills it with the received message. It then builds a send packet with the ICC buffer number and sends the packet to the correct destination process, which is determined from the first word of the

input data buffer. If the message is a status message, it sets the packet code to STAMSG. Otherwise, it sets the code to ICCMSG.

### 7.2 Macro Modules

*Subroutine COM_READ*

COM_READ is an AST routine that executes when a message is received by the Ethernet. It passes data to WRTMSG for transmission to the correct destination process and checks that all messages from the other system have been processed. It then reposts a read QIO to the Ethernet and resets the read-timeout timer.

*Subroutine INIT_PRI*

This subroutine runs on the primary system CPU. It attempts to initialize communications with the other system. A read QIO is posted to the Ethernet. Then a "set normal response mode" (SNRM) message is sent to the other system and a 1 second timer is set. A wait for the read QIO event flag or the timer event flag is set. Upon the completion of the wait for logical OR system service, the read event flag is checked. If it is not set, another write message and timer cycle is started. If it is set, the read QIO completed and the circuit with the other system is initialized. A line-recovered message is sent to SYSMON, a keep-alive timer is started, a read QIO is posted with a time-out timer, and a write attention QIO with AST routine RCVMSG is set to the ICC mailbox.

*Subroutine INIT_SEC*

This routine runs on the secondary system CPU. A read QIO is posted to the Ethernet channel. Upon completion, the data are checked for an SNRM message. If none is present, another read is posted. Otherwise, the circuit is initialized. A keep-alive message is sent, a read QIO is posted with a time-out timer, and a write attention QIO with AST routine RCVMSG is set on the ICC mailbox.

*Subroutine KA_TMO*

This is an AST routine that executes every 2 seconds when the keep-alive timer completes. The system status is stored in an Ethernet buffer and sent to the other system. Then the keep-alive timer is restarted.

*Subroutine RCVMSG*

This is an AST routine that executes when a message is sent to the ICC mailbox. It checks for a valid message. The address of the ICC buffer is passed to SEND_FRAME for transmission to the other system. Then the buffer is deallocated. If an event flag is requested, it is set. The cycle is repeated until the ICC mailbox is empty. Then another write attention is set.

*Subroutine READ_TMO*

This AST routine is entered if the read time-out timer completes. The communication line is marked down; then a failure message is sent to the operator, and XMTEF is set to let ICC know to attempt to reinitialize the ICC procedure as a primary system procedure.

### 7.3 Hardware Components

*7.3.1 Ethernet Controller*

The Ethernet controller is embedded on the CPU card of the VAX 4200. It has both standard transceiver and thinwire connectors on the front panel of the CPU module. In the DSPR system, the thinwire connection is used.

## 8. SYSTEM TIMING COMPONENTS

This procedure maintains the DSPR system timing accuracy and provides time stamping for data sent to the NSSC. The system time has both a hardware and software component. The year, month, day, hour, minute, and second are stored as integer variables in the system database, global common area DSPCOM, and are updated once per second by the software. This software clock is initialized by reading a Hewlett-Packard 59309A HP-IB digital clock (to obtain the months, days, hours, minutes, and seconds) and the VAX 4200 CPU's internal clock (to obtain the year). The software clock is updated once per second by the time stamp procedure, which controls a KWV11C programmable real-time clock. This clock refines the timing resolution to 0.1 ms. A Hewlett-Packard 5061B cesium beam frequency standard provides precise 5 MHz and 1 pulse-per-second (pps) signals. The method of obtaining timing information is to set up to read the digital clock, wait for a pulse from the cesium clock to occur, then read the digital clock.

### 8.1 Fortran Modules

*Subroutine RD_HPCLK*

This routine places the system date and time into array SYS_TIME, which is later transferred to array RFTIME in global common DSPCOM.

*Subroutine SNDMSG*

SNDMSG sends an error code to SYSMON.

### 8.2 Macro Modules

*Program CLOCK*

This is the driver program that controls the timing for the entire DSPR system. It maintains the software component of the system time by continuously updating values in array RFTIME in the system database. These values are used for time stamping all data.

*Subroutine CLK_ERR*

This module contains the asynchronous clock error handling routine.

*Subroutine INIT_TIME*

INIT_TIME reads the VAX system time to obtain the current year and stores it in array SYS_TIME.

*Subroutine KWV_CIN*

KWV_CIN contains KWV "connect to interrupt" routines that obtain the once-per-second time update for the KWV11C from the cesium beam clock. The routines are the following:

KWV_START starts the KWV11;

KWV_ISR is an interrupt service routine;

KWV_CNCL cancels an I/O operation in progress and turns off the KWV11;

KWV_INIT initializes the size of the buffer and the beginning address of the buffer, initializes the KWV routine names, stores the channel number and user AST address, and performs the connect-to-interrupt.

*Subroutine MAP_IO_PAGE*

This subroutine maps to the input/output page.

*Subroutine UPDATE_TIME*

The routine moves the data in array SYS_TIME to array RFTIME when the KWV11C is interrupted by the 1 pps clock.

## 8.3 Hardware Components

### *8.3.1 KWV11C Programmable Clock*

The KWV11C is a programmable real-time clock used to determine time intervals or to count events. The counter is a 16 bit divider and can be driven by one of five internal base frequencies: 1 MHz, 100 kHz, 10 kHz, 1 kHz, and 100 Hz. It has a Schmitt trigger input which can be used to reset the counter. In the DSPR system, the 10 kHz base frequency is used for the input to the counter. Therefore, the counter gives a 0.1 ms resolution referenced to the second. The KWV11Cs were modified so that the counter value could be read at any time through the preset/buffer register. (See *AXV11-C/KWV11-C Analog Module and Real-Time Clock Module User's Guide.*)

### *8.3.2 Hewlett-Packard 5061B Cesium Beam Frequency Standard*

The HP 5061B is an atomic resonance device that provides a very high accuracy ($\pm 3 \times 10^{-12}$) primary frequency standard. It is a compact, self-contained unit that uses a cesium beam tube resonator to stabilize the output frequency of a quartz crystal oscillator. The crystal oscillator is divided down to produce frequencies of 5 MHz, 1 MHz, 100 kHz, and 1 pps. All four frequencies are available at both the front and rear panels. The 5 MHz, 1 MHz, and 100 kHz output levels are 1 VRMS (minimum) into a 50 ohm load and the 1 pps output level is a 10 volt, 20 microsecond pulse width into a 50 ohm load of a digital clock input. (See *5061B Cesium Beam Frequency Standard, Operating and Service Manual.*)

### *8.3.3 Hewlett-Packard 59309A HP-IB Digital Clock*

The HP 59309A Digital Clock operates as a 24 hour clock that displays the month, day, hour, minute, and second. Upon command, it outputs time via the interface bus. It also provides an ASCII

character format for an IEEE 488 bus interface connector for use in other parts of the DSPR. The front panel provides a visual display of the month, day, hour, minute, and second for the operator. This is used primarily when first setting the correct time. The clock is driven by the 1 pps signal from the cesium beam frequency standard. (See *59309A Operator Service Manual.*)

## 9. OPERATIONAL TESTS

The Operational Tests, also called OPTESTs, are on-line confidence tests used to exercise one or more components of the DSPR system. They give indications about the performance of the system or of critical components of the system. All modules are written in Fortran. Hardware used by the operational tests includes an IEQ11 IEEE 488 Bus Controller, a Hewlett-Packard 8657A Signal Generator/Calibrator, and a Hewlett-Packard 5087A Distribution Amplifier.

### 9.1 Fortran Modules Common to Operational Tests

*Program OPT*

This is the driver program for all of the operational tests, except for the Activity Monitor, which is incorporated into the system monitor dynamic activity display. It checks the initialization send packet and ICC buffer for the correct values. It changes the state of the system or CPU to idle if required and sends a log message to SYSMON signalling OPTEST commencing. Then it calls the appropriate subroutine to handle the specific test being executed. On return from the subroutine, an "OPTEST completed" log message is sent to SYSMON, and the system and CPU states are restored. If the test is running in the primary system and both are requested, the ICC initialization buffer will be forwarded to the secondary system via ICC; otherwise, the initialization buffer will be deallocated.

*Subroutine CALSET*

CALSET sets up the calibrator. This module does the set-up processing to make it possible for the Alert and System Signal Confirmation tests to access the calibrator.

*Subroutine CALSHUT*

This subroutine halts the calibrator. It is called during the Alert or System Signal Confirmation tests.

### 9.2 Fortran Modules for Alert Sensitivity Operational Test (OPALRT)

*Subroutine OPALRT (INITBF)*

This module turns the calibrator on at the appropriate frequency and drives the subroutines to collect and analyze the data. Each step is 4 seconds long on the primary system and 64 seconds long on the secondary system. The calibrator is turned "off" for an equal period of time between steps by setting it to its lowest level (-136 dBm). (INITBF is the number of the ICC buffer that has been allocated by program OPT.)

*Subroutine ALLOC_DF (ID)*

This module is used to allocate one digital filter for the primary system or digital filter 3, and one other for the secondary system to run OPALRT. If no digital filter can be found in the allotted time, the program exits. Digital filters are allocated by setting their status variable to 2, so the target select

procedure will not use them. ALLOC_DF reactivates the CPU from an idle state. (ID is the number of the digital filter that is allocated.)

*Subroutine ALRT_MSG (AMP)*

This module reports percentage hits for OPALRT. It calculates the percentage of hits during an OPALRT step and puts the results in an ICC buffer in ASCII format. (AMP is the signal amplitude for the calibration step.)

*Subroutine COLECT (ID)*

This module starts up data collection while the calibrator is on. It also puts the calibration frequency on the secondary system's elimination lists. (ID is the number of the digital filter on which to collect data.)

### 9.3 Fortran Modules for Digital Filter Operational Test (OPDFT)

*Subroutine OPDFT (INITBF)*

This module performs the digital filter operational test. After waiting for all data processing to finish, data are collected on all digital filters and compared for any discrepancies greater than the allowed bit tolerance. If there are any, the test is repeated once. (INITBF is the ICC initialization buffer number passed to OPDFT by OPT.)

*Subroutine DFCOL (FREQT)*

This module tunes the digital filters and initiates collection on all three at once. If any digital filters time out, subroutine DFTMO is called to mark them down. (FREQT is the frequency to which the digital filters are tuned.)

*Subroutine DFTMO (NUM)*

Module DFTMO is called when a digital filter has timed out. If the digital filter has not been previously marked "down," an error message is sent to SYSMON. Otherwise, a return is executed. (NUM is the number of the digital filter that timed out.)

### 9.4 Fortran Modules for System Signal Confirmation Operational Test (OPSYS)

*Subroutine OPSYS (INITBF)*

OPSYS performs the system signal confirmation test. It waits for DP to finish, turns on the calibrator, and collects data. The data are processed by subroutine PHCALC. The results from PHCALC are placed in ICC buffers and shipped to SYSMON in the primary system. (INITBF is the ICC initialization buffer that was passed to OPSYS by OPT.)

*Subroutine PHCALC (IDF, DAT)*

This subroutine is used by task OPSYS to compute the average phase of each antenna relative to antenna 4. The values obtained by 10*log10 of the amplitudes (squared) are also computed and are left

in the data arrays in place of the real portion of the complex numbers. (IDF is the number of the digital filter on which the data were collected, and DAT is the array that holds the results of the computations.)

## 9.5 Fortran Modules for RF Calibration Operational Test (RFCAL)

*Subroutine RFCAL*

This module performs the reference calibration test. It determines the first available digital filter. If a digital filter is available, the routine sets the frequency of the digital filter to CAFREQ and marks the digital filter in use. If a digital filter is not available at first, the routine tries once per second for 4 seconds to obtain a digital filter. If a digital filter is not available during this time interval, the routine reports a digital filter time-out error message.

## 9.6 Hardware Components

### *9.6.1 IEQ11 IEEE 488 Bus Controller*

The IEQ11 is a DEC device that interfaces between the Qbus and the General Purpose Interface Bus (GPIB), which conforms to the IEEE-488 standard. It contains two independent GPIB controllers, only one of which is used in the DSPR system. Each channel can support 14 devices plus the controller itself. In the DSPR system, the devices on the GPIB are the HP 8657A signal generator and the HP 59309A digital time of year clock.

### *9.6.2 Hewlett-Packard 8657A Signal Generator/Calibrator*

The HP 8657A is a programmable synthesized signal generator with a frequency range of 0.1 to 1040 MHz and an amplitude range of +13 dBm to -143.5 dBm. It is programmable through an HP-IB (Hewlett Packard's implementation of IEEE Standard 488) interface. It is used by the operational tests to inject a signal of known frequency and amplitude into the antenna pre-amplifiers. The signal cables from the HP 8657A to the individual pre-amps are cut to the same length so that the signal phases at the inputs to the pre-amps are identical. (See *HP 8656B/8657A/8657A Synthesized Signal Generator, Operation and Calibration Manual.*)

### *9.6.3 Hewlett-Packard 5087A Distribution Amplifier*

The HP 5087A Distribution Amplifier is a three-input channel, four-outputs-per-channel device. The DSPR uses only the 5 MHz input channel; of the four outputs, only two are used. This device buffers and isolates the 5 MHz reference signal from the cesium beam frequency standard for the Hewlett-Packard 8657A calibrator and for the RF subsystem local oscillator generator. (See *Operator Service Manual for 5087A Distribution Amplifier.*)

# 10. UTILITY BUS CONTROL

This subsystem, controlled by procedure UTIL_BUS, directs and supervises activity on the utility bus and related hardware elements that were designed and developed by NRL. It executes two basic functions: performing system requests for utility bus operations and monitoring activity on the utility bus.

The utility bus is used to initialize the custom hardware in the DSPR system and to detect changes in the hardware during system operation. The hardware devices on the utility bus are daisy-chained, and each has a specific address. The bus is 16 bits wide with the eight most significant bits used for

addressing and the lower eight bits used for data. The devices on the bus are the spare channel and local oscillator exchange cards, the data distribution and test card, and the command status module (CSM). The phone relay card is controlled through the CSM. A prototype digital delay device is installed at the Hawkinsville receiver station. The utility bus interfaces to the VAX 4200 through a DRV1W interface.

## 10.1 Fortran Modules

*Program UTIL_BUS*

UTIL_BUS uses a DRV1W interface device to control communication between the VAX 4200 and the various utility cards. Each CPU has its own utility bus controller that communicates to specific parts of the DSPR hardware. The cards attached to the utility bus send hardware status information to SYSMON and receive hardware commands sent from SYSMON.

*Subroutine CK_INTERRUPT*

CK_INTERRUPT checks the utility bus for an interrupt. If there is an interrupt, it reads the bus to determine which card made the interrupt. If there is a change in the bit pattern between the data previously read from that card and the current data, as a result of the interrupt, then it passes that information to SYSMON.

*Subroutine CLR_INTERRUPT*

This routine waits for a clear-interrupt bit before returning back to UB_INIT. It prevents UB_INIT from starting its initialization process when there is an interrupt on the utility bus.

*Subroutine READ_CARD (IOWORD)*

READ_CARD performs a read on one of the utility cards attached to the utility bus via the DRV1W. Argument IOWORD, on input, contains the address of the card to be read. As an output argument, it contains the address from which information came, plus data from the card.

*Subroutine UB_INIT*

This routine initializes the DRV1W utility bus system. It makes sure that the interrupt bit is clear by checking its status in the control status register, turns off and clears data delay (if applicable), reads the various cards for their present values, and places the values in array IOWORD. It then calls subroutine BIT_DIFF to determine the bit changes from the time the cards were previously read, and sends the bit change information to SYSMON.

*Subroutine WRITE_CARD (OUTWORD)*

WRITE_CARD performs a write to one of the utility cards attached to the utility bus. OUTWORD contains the address of the card to write to and the data to be passed to it.

## 10.2 Macro Modules

*Subroutine BIT_DIFF*

This routine determines if a bit has changed status from the previous time the utility card had input information. A table is maintained of the bit patterns for each utility card (UB_IN). For each of the eight data bits for a utility card, the routine determines the status difference between the new and the old. A logical array named CHANGE is dimensioned to 8 and is used to store differences. Each element in the array corresponds to a bit position: element 1 to bit 0, element 2 to bit 1, element 3 to bit 2, etc. Each element of the array has a numeric value determined as follows:

If no change, then a value of 0.
If a change of 0 to 1, then a value of 1.
If a change of 1 to 0, then a value of 2.

This subroutine also updates the table entry for the utility card (UB_IN) to reflect the latest bit pattern. Argument OLD_CARD contains the old value of utility card and NEW_CARD receives the new value. The Fortran call is CALL BIT_DIFF (OLD_CARD, NEW_CARD, CHANGE).

*Subroutine CHANGE_BITS*

This routine modifies specific bits in the low byte of the IOWORD variable passed from UTIL_BUS. The bits that are set in the low byte of IOWORD are in the variable BITSET. The bits that are cleared in the low byte of IOWORD are in the variable BITSCL. The final IOWORD sent from this routine to UTIL_BUS contains the address of the utility card in the high byte and commands to send to the utility card in the low byte. The Fortran call is CALL CHANGE_BITS (BITSET, BITSCL, IOWORD).

## 10.3 Hardware Components

*10.3.1 DRV1W Interface*

The DRV1W is discussed under Interferometer Data Collection.

*10.3.2 Phone Line Relay Card*

The phone line relay card is a custom-built device that controls the interface between the modem of each system and the common phone line to the NAVSPASUR Processing and Operations Center. Only the modem on the primary system is connected to the phone line.

# 11. MISCELLANEOUS ROUTINES

Routines described in this section are general-purpose subroutines that are called by various procedures of the DSPR software.

## 11.1 Fortran Modules

*Subroutine ASCNUM (N, A)*

This routine converts an INTEGER*2 to three ASCII characters. No error checking is performed to determine if the integer is in the proper range between 0 and 999.

*Subroutine CLRS*

This routine clears the terminal screen and leaves the cursor at the top of the screen.

*Subroutine ETEXT (LUN, STRING, LINE, COLMN)*

This routine prints a error text string specified by STRING on the logical unit specified by LUN. The LINE and COLMN arguments control where the text string is printed on a video terminal screen. Synchronous I/O is used.

*Subroutine INIT_OPTBUF (SNDTYP, ICCTYP, N)*

This module allocates and initializes an ICC buffer to be sent to SYSMON. It initializes the buffer with the destination code and the type of ICC message. It also initializes the OPTPKT send packet with the type of send packet and the ICC buffer number. Argument SNDTYP is the type of send packet being set; ICCTYP is the type of ICC message being sent; and N is the ICC buffer number.

*Subroutine LOCK*

This routine locks global common area DSPCOM to permit a series of operations to be made in an uninterrupted fashion. It can be used to coordinate serial access to the DSPR system common area by multiple concurrent tasks. It attempts to lock DSPCOM, and if successful, returns to the caller. If DSPCOM is already locked by another task, it waits for that task to unlock DSPCOM then re-locks DSPCOM for the calling task and returns.

*Subroutine SCLR*

This routine clears the terminal screen and leaves the cursor at the top of the screen.

*Subroutine TEXT (LUN, STRING, LINE, COLMN)*

This routine prints a text string specified by STRING on the logical unit specified by LUN. The LINE and COLMN arguments control where the text string is printed on a video terminal screen. Synchronous I/O is used.

*Subroutine TRACE (CHAN, COUNT, STRING)*

This module constructs a trace message using the ASCII string STRING of byte count COUNT. It places it in a send packet as embedded text, with destination SYSMON and with error code TR01 to produce a trace log message. The packet is then sent to CHAN. If the byte count is greater than 19, it is truncated to 19, since that is the limit for the length of embedded text in the send packet.

*Subroutine UNLOCK*

This routine unlocks DSPCOM for use by other tasks.

## 11.2 Macro Modules

*Subroutine ALLOC_ICCBUF*

This routine allocates the first available interprocessor message buffer (BUFNUM) for the caller. If no buffers are available and input argument is zero or positive, it waits until one is available. If no buffers are available and input argument is negative, it returns immediately with an output argument equal to the input argument.

*Subroutine CVT_TIME*

This routine converts the DSPR system time and date found in the DSPCOM variable called RFTIME to a form of *dd-mmm-yyyy hh:mm:ss.cc*. This converted time is output to the variable ARRAY. The Fortran call for this routine is CALL CVT_TIME (ARRAY).

*Subroutine CVT_TS*

This routine converts the current time and date array of the form *yyyy,mm,dd,hh,mm,ss,ms* to a form of *dd-mmm-yyyy hh:mm:ss.cc*. The converted time is output to the variable ARRAY. The Fortran call is CALL CVT_TS (IN_ARRAY, OUT_ARRAY).

*Subroutine DAT*

This routine converts the DSPR system date found in DSPCOM variable RFTIME to a form of *dd-mmm-yyyy* and outputs it to the variable ARRAY. The Fortran call is CALL DAT (ARRAY).

*Subroutine DEALLOC_ICCBUF*

This routine deallocates the interprocessor message buffer specified by BUFNUM. The Fortran call is CALL DEALLOC_ICCBUF (BUFNUM).

*Subroutine FILL_STATUS*

This routine fills the interprocessor message buffer ICCBUF specified by N with the DSPR system status from global common area DSPCOM. The Fortran call is CALL FILL_STATUS (N, MODE).

*Subroutine INIT_TASK*

This routine creates a detached process and starts a task running in that process. The Fortran call is CALL INIT_TASK (IMAGE, PID_ADD). Argument IMAGE indicates the task to start; PID_ADD is the address of the process id.

*Subroutine RECEIVE*

This routine replaces the PDP-11 RCVD$ executive directive that received a 13 word block of data from another task. RECEIVE can be called by any DSPR task to receive data from another task in a 15 word buffer format. Virtual mailboxes are used to pass the data between the tasks. The Fortran call is CALL RECEIVE (MBX_CHAN, BUFFER, RCVEF, IOSB).

*Subroutine REC_OR_EX*

REC_OR_EX attempts to read an initialization packet from the mailbox. If unsuccessful, the procedure exits. The Fortran call is CALL REC_OR_EX (MBOX, AST_ADD).

*Subroutine SEND*

This routine replaces the PDP-11 RSX executive directive SDAT$ that passed a 13 word block of data to another task. SEND can be called by any DSPR task to send data to another task in a 15 word buffer format. Virtual mailboxes are used to send the data to the other task. The Fortran call is CALL SEND (MBX_CHAN, BUFFER, SENDEF, IOSB).

*Subroutine SET_WRTATTN*

This routine sets a write attention to a mailbox. The Fortran call is CALL SET_WRTATTN (CHAN, AST_ADD).

*Subroutine TIM*

WAIT converts system time, RFTIME, into time in the format hh:mm:ss, which is output to variable, ARRAY. The Fortran call is CALL TIM (ARRAY).

*Subroutine WAIT*

This routine sets up a timer and waits for a specified number of seconds. It is an emulation of the PDP-11 RSX system directive MRKT$. The Fortran call is CALL WAIT (%val(EFN), DAYTIM). If the DAYTIM that is needed is not processed by INIT_TIMER.MAR, then the time value must be set up by using VMS system directive SYS$BINTIM.

## ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| **ADC** | Antenna data collection |
| **AP** | Array processor |
| **APIO** | Array processor input/output |
| **AST** | Asynchronous trap |
| **CCITT** | Comité Consultatif Internationale de Télégraphique et Téléphonique |
| **CSM** | Command status module |
| **dBm** | Decibels above 1 milliwatt |
| **DD&TC** | Data distribution and test card |
| **DEC** | Digital Equipment Corporation |
| **DF** | Digital filter |
| **DFT** | Discrete Fourier transform |
| **DIO** | Direct input/output |
| **DLC** | Data line communication |
| **DMA** | Direct memory access |
| **DP** | Data processing |
| **DSPR** | Digital Signal Processing Receiver |
| **DTP** | Data transmission packet |
| **FFT** | Fast Fourier transform |
| **GPIB** | General purpose interface bus |
| **Hz** | Hertz |
| **ICC** | Interprocessor communications |
| **kb/s** | Kilobytes per second |
| **kHz** | Kilohertz |
| **MAP** | MAP 4000 Array Processor |
| **MB** | Megabyte |
| **Mflop** | Megaflop |
| **MHz** | Megahertz |
| **MIPS** | Million instructions per second |
| **ms** | Millisecond |
| **NAVSPASUR** | Naval Space Surveillance |
| **NRL** | Naval Research Laboratory |
| **NSSC** | Naval Space Surveillance Center |
| **OPALRT** | Alert sensitivity operational test |
| **OPDFT** | Digital filter operational test |
| **OPSYS** | System signal confirmation operational test |
| **OPTEST** | Operational test |
| **pps** | Pulse per second |
| **RF** | Radio frequency |
| **RFCAL** | RF calibration operational test |
| **SDLC** | Synchronous data line control |
| **SNRM** | Set normal response mode |
| **SPADATS** | Space Detection and Tracking System |
| **SYSMON** | System monitor and control |

| | |
|---|---|
| **TCB** | Transport control buffer |
| **TRGSEL** | Target selection |
| **UTIL_BUS** | Utility bus control subsystem |
| **VRMS** | Volts root-mean-squared |

## REFERENCES

Manuals Published by Digital Equipment Corporation, Maynard, Massachusetts:

*AXV11-C/KWV11-C Analog Module and Real-Time Clock Module User's Guide*, DEC document EK-AXVAA-UG-001, 1986.

*DRV11-WA General Purpose DMA Interface User's Guide*, DEC document EK-DRVWA-UG-002, 1986.

*DSV11 Communications Option Technical Description*, DEC document EK-DSV11-TD-001, 1989.

*DZQ11 Asynchronous Multiplexer User's Guide*, DEC document EK-DZQ11-UG-001, 1984.

*Installing and Using the VT420 Video Terminal*, DEC document EK-VT420-UU-002, 1990.

*LA120 User Guide*, DEC document EK-LA120-UG-005, 1984.

*VAX Wide Area Network Device Drivers, Installation Guide Specifications, and Programmer's Guide*, DEC document AA-LN26B-TE, 1989.

Manuals Published by Hewlett-Packard Company, Spokane, Washington:

*5061B Cesium Beam Frequency Standard, Operating and Service Manual*, manual number 05061-9078, 1986.

*59309A Operator Service Manual*, manual number 59309-90004.

*HP 8656B/8657A/8657B Synthesized Signal Generator, Operation and Calibration Manual*, manual number 08657-90115, 1991.

*Operator Service Manual for 5087A Distribution Amplifier*, manual number 05087-90008.

Other Manuals:

*MAP 4000 Installation and Diagnostics*, *MAP 4000 Software Utilities*, *MAP 4000 Scientific Subroutine Library*, and *MAP 4000 DIO1 User Manual*, CSP Inc., Billerica, Massachusetts, 1990.

*V.3224/V.3225 Installation and Operation*, manual number 5-209-5173-02-014, Universal Data Systems, Inc., Huntsville, Alabama, 1990.